

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A method of detecting unauthorized executable programs ~~code~~ resident in a computer system ~~memory~~ comprising the steps of:

~~a) receiving a trusted hash value representative of a hash value for generation by a predetermined hashing process of predetermined data stored in memory within the computer system for a trusted state of executable programs in execution within the computer system if an unauthorized executable program is other than resident in the computer system;~~

b) hashing first data stored in data storage ~~memory~~ within the computer system using a selected ~~the predetermined~~ hashing process to determine a computed hash value, wherein the first data includes data representing for a current state of at least one application executing executable programs in execution within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a known state of the one or more applications executing in the computer system, wherein the second data includes a system memory location indicative of the at least one application executing within the computer system; and

e) comparing the computed hash value with ~~and~~ the trusted hash value to determine whether there is unauthorized executable code in the computer system ~~differences between the data and the predetermined data.~~

2. (Currently Amended) The method of claim 1, further ~~A method of detecting unauthorized executable programs resident in a computer system memory according to claim 1 including the steps of comprising:~~

~~aa)~~ receiving user authorization information;

~~aaa)~~ authenticating the user authorization information to perform at least one of authorize and identify a user; and

~~aaaa)~~ when the user is at least one of authorized or identified, requesting security data of the user.

3. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ of claim 2, wherein the authorization data is at least a biometric information sample₁[[;]] and wherein the ~~step of~~ authenticating includes a ~~step of~~ comparing the at least a biometric information sample to a previously stored biometric template.

4. (Currently Amended) ~~A The method of detecting unauthorized executable programs resident in a computer system according to~~ of claim 2, further comprising the steps of:

when the comparison is indicative of other than ~~an~~ unauthorized executable code program resident in a computer system, providing the requested security data relating to the user.

5. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ of claim 1, further comprising the steps of:

receiving a request for security data from an application in execution in the computer system; and,

when the comparison is indicative of other than an unauthorized executable program[[s]] resident in a computer system, providing security data to the application.

6. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ of claim 8, wherein the trusted hash value and the computed hash value are determined by a same trusted security application executing locally on a processor of a same computer system at different times, the trusted hash value determined when the ~~computer system is in a~~ known state is a secure state.

7. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 6, wherein the trusted hash value is digitally signed.

8. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 7, wherein the hashing data includes ~~ing the step of~~:

b1) verifying an authenticity of the digitally signed trusted hash value.

9. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 8, further comprising the steps of:
receiving a request for security data from an application in execution in the computer system; and,

when the authenticity of the digitally signed trusted hash value is verified and the comparison is indicative of other than ~~an~~ unauthorized executable code ~~programs~~ resident in a computer system, providing the requested security data to the application.

10. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 9, wherein the application and the predetermined hashing process are both executed on a same processor of the computer system.

11. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 8, comprising the step of:

d) when the computed hash value and the trusted hash value are other than indicative of a ~~known~~ secure state, issuing a notification that ~~an~~ unauthorized executable code ~~program~~ is detected within the computer system.

12. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 11, further comprising the step of:

e) when the computed hash value and the trusted hash value are other than indicative of a ~~known~~ secure state, preventing access to the computer system.

13. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 7, further comprising: the step of

transmitting the trusted hash value to a second other computer system in communication with the computer system and retrievably storing the computed hash value within the second other computer system.

14. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 13, further comprising including the step of transmitting the computed hash value to the second other computer system for comparison with the trusted hash value by a processor of the second other computer system.

15. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 14, wherein the computed hash value is ~~a value~~ determined in dependence upon the predetermined data existing in memory within the computer system and some time dependent data of the computer system.

16. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to~~ claim 14, wherein the second other computer system includes a trusted source and wherein security data is stored for provision to applications in execution on systems that are known to be secure.

17. (Currently Amended) A method of detecting unauthorized executable ~~programs~~ code resident in a computer system comprising ~~the steps of~~:

a) providing a trusted security application executable on a processor of the computer system for determining a hash value using a ~~predetermined selected~~ hashing process applied to of predetermined data existing in memory within the computer system, wherein the predetermined data includes system memory locations indicative of executable programs in operation;

b) ~~hashing the predetermined selected~~ data existing in memory within the computer system using the predetermined process to determine a hash value;

e) digitally signing the hash value to provide a trusted hash value; and

d) retrievably storing the trusted hash value, ~~wherein the hash value is determined absent an unauthorized executable program being present within the computer system~~[[.]],
and

wherein the predetermined data relates to programs in execution on the processor of the computer system when the computer system is in a ~~known~~ secure state.

18. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to claim 17,~~ further comprising the steps of:

e) comparing a computed hash value with the trusted hash value to detect changes to the predetermined data existing in memory within the computer system.

19. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to claim 18,~~ further comprising the step f) verifying the authenticity of the digital signature of the trusted hash value.

20. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to claim 19,~~ further comprising the step of:

g) ~~when the computed hash value and the trusted hash value are indicative of a same trusted state of a computer system,~~ providing security data from a trusted source to an application in execution on the system.

21. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to claim 20,~~ further comprising the step of:

h) ~~when the computed hash value and the trusted hash value are other than indicative of a same secure state of the system,~~ sending a notification ~~to a system administrator.~~

22. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to claim 1,~~ wherein the predetermined data includes DLL tables.

23. (Canceled)

24. (Currently Amended) ~~The A method of detecting unauthorized executable programs resident in a computer system according to claim 1,~~ wherein the predetermined data is hashed in an absolute memory location independent fashion.

25. (New) The method of claim 1, wherein if the computed hash value and the trusted hash value compare within a selected limit, there is no unauthorized executable code in the computer system.

26. (New) The method of claim 1, further comprising performing a user authorization process for verifying that a user is authorized.

27. (New). The method of claim 26, wherein the one or more applications executing in the computer system includes at least one untrusted application and at least one trusted application, and further comprising transmitting a password request from the at least one untrusted application to the at least one trusted application.

28. (New) The method of claim 27, wherein the transmitting a password request from the at least one untrusted application to the at least one trusted application is in response to a user's attempt to access a data file associated with the at least one untrusted application.

29. (New) The method of claim 27, wherein the user authorization process comprises:
detecting the password request from the at least one untrusted application by the at least one trusted application;
prompting the user to input authorization information; and,
comparing the input authorization information with information retrieved from the at least one trusted application, wherein if the input authorization information successfully compares with the information retrieved from the at least one trusted application, the user is an authorized user.

30. (New) The method of claim 29, wherein the hashing data, retrieving a trusted hash value and the comparing the computed hash value with the trusted hash value are carried out if the input authorization information successfully compares with the information retrieved from the at least one trusted application.

31. (New) The method of claim 29, wherein the input authorization information comprises at least biometric information, and wherein the comparing includes comparing the at least biometric information with a previously stored biometric template.

32. (New) The method of claim 29, wherein the at least one trusted application includes a user verification database, and wherein the input authorization information is compared with information retrieved from the user verification database.

33. (New) The method of claim 1, wherein the at least one trusted application includes a hash generator and wherein the hashing data is carried out in the hash generator.

34. (New) The method of claim 1, wherein the trusted hash value is encrypted.

35. (New) The method of claim 34, wherein the trusted hash value is digitally signed.

36. (New) The method of claim 34, further comprising:
decrypting the encrypted trusted hash value; and,
comparing the decrypted trusted hash value with the computed hash value, wherein if the computed hash value and the decrypted trusted hash value compare within a selected limit, there is no unauthorized executable code in the computer system.

37. (New) The method of claim 35, further comprising:
decrypting the digitally-signed trusted hash value; and,
comparing the decrypted trusted hash value with the computed hash value wherein if the computed hash value and the decrypted trusted hash value compare within a selected limit, there is no unauthorized executable code in the computer system.

38. (New) The method of claim 27, wherein the trusted hash value is digitally signed and further comprising:

decrypting the digitally-signed trusted hash value;
comparing the decrypted trusted hash value with the computed hash value; and,
refusing the password request from the at least one untrusted application if the computed hash value and the decrypted trusted hash value do not compare within a selected limit.

39. (New) The method of claim 1, wherein the data storage comprises at least a volatile memory.

40. (New) The method of claim 1, wherein the data storage comprises at least a disk drive.

41. (New) The method of claim 1, further comprising:
determining that the computer system is in a known state;
hashing data representing the known state of the at least the one application executing in the computer system using the selected hashing process to create the trusted hash value;
and,
encrypting the trusted hash value.

42. (New) The method of claim 41, further comprising:
retrievably storing the trusted hash value in the data storage

43. (New) The method of claim 41, wherein the determining comprises:
performing a user authorization process to determine an authorized user, wherein
and,
receiving a command from the authorized user that the computer system is in a known state.

44. (New) The method of claim 41, wherein the trusted hash value is digitally signed.

45. (New) The method of claim 34, wherein the computer system includes a plurality of networked computers, and wherein the encrypted trusted hash value is stored in a secure one of said plurality of computers, the method further comprising:

receiving in the secure computer, the computed hash value transmitted from at least a first computer;

decrypting the encrypted trusted hash value in the secure computer;

wherein the comparing of the decrypted trusted hash value with the computed hash value occurs in the secure computer; and,

if the computed hash value and the trusted hash value compare within a selected limit:

retrieving a password from a memory in the secure computer; and,

transmitting the retrieved password to the at least a first computer.

46. (New) The method of claim 45, wherein the one or more applications executing in the computer system includes at least one untrusted application executing on the at least a first computer and at least one trusted application executing on the at least a first computer, the method further comprising:

detecting a password request from the at least one untrusted application by the at least one trusted application;

prompting the user to input authorization information; and,

comparing the input authorization information with information retrieved from the at least one trusted application; and,

wherein if the input authorization information successfully compares with the information retrieved from the at least one trusted application, the user is an authorized user.

47. (New) The method of claim 45, wherein the trusted hash value is digitally signed, and further comprising:

decrypting the digitally-signed trusted hash value in the secure computer; and,

comparing the decrypted trusted hash value with the computed hash value in the secure computer, wherein if the computed hash value and the decrypted trusted hash value

compare within a selected limit, there is no unauthorized executable code in the at least a first computer.

48. (New) The method of claim 47, further comprising:
refusing the password request from the at least one untrusted application if the computed hash value and the decrypted trusted hash value do not compare within a selected limit.

49. (New) The method of claim 45, further comprising:
determining that the at least a first computer is in a known state;
hashing data representing the known state of the at least one application executing in the at least a first computer using the selected hashing process to create the trusted hash value; and,
encrypting the trusted hash value.

50. (New) The method of claim 49, further comprising:
transmitting the encrypted trusted hash value to the secure computer; and
storing the encrypted trusted hash value in the secure computer.

51. (New) The method of claim 45, further comprising:
encrypting the computed hash value in the at least a first computer prior to transmission; and,
decrypting the computer hash value in the secure computer.

52. (New) The method of claim 45, wherein the retrieved password is encrypted, and further comprising:
decrypting the retrieved password in the at least a first computer.

53. (New) The method of claim 45, further comprising:

transmitting an incorrect password to the at least one untrusted application of the at least first computer if the computed hash value and the trusted hash value do not compare within a selected limit:

54. (New) The method of claim 45, further comprising
transmitting a lock command to the at least one untrusted application of the at least first computer if the computed hash value and the trusted hash value do not compare within a selected limit:

55. (New) The method of claim 2, further comprising:
prompting a user to verify that the unauthorized executable code is from a known source if the computed hash value and the trusted hash value do not compare within a selected limit value.

56. (New) The method of claim 45, wherein if the computed hash value and the trusted hash value do not compare within a selected limit, there is unauthorized executable code in the at least a first computer, and further comprising:
prompting a user to verify that the unauthorized executable code is from a known source.

57. (New) The method of claim 1, wherein the known state is a secure state.

58. (New) The method of claim 1, wherein the known state is an initial state of an operating system within the computer system.

59. (New) The method of claim 45, wherein the known state is a secure state.

60. (New) The method of claim 45, wherein the known state is an initial state of an operating system within the computer system.

61. (New) A system for detecting unauthorized executable resident in a computer system, the system comprising a computer processor programmed to perform the method comprising:

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a known state of the one or more applications executing in the computer system, wherein the second data includes data from at least a system memory location indicative of the at least one application executing within the computer system; and

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system.